On learning matrices with orthogonal columns or disjoint supports

Kevin Vervier[1,2,3], Pierre Mahé[1], Alexandre D'Aspremont[4], Jean-Baptiste Veyrieras[1], and Jean-Philippe Vert[2,3]

[1] Data and Knowledge Lab, Biomerieux, 69280 Marcy l'Etoile, France
[2] Centre for Computational Biology, Mines ParisTech, 77300 Fontainebleau, France
[3] Institut Curie, INSERM U900, 75005 Paris, France
[4] CNRS and D.I. UMR 8548, Ecole normale supérieure, 75005 Paris, France

Abstract. We investigate new matrix penalties to jointly learn linear models with orthogonality constraints, generalizing the work of Xiao et al. [24] who proposed a strictly convex matrix norm for orthogonal transfer. We show that this norm converges to a particular atomic norm when its convexity parameter decreases, leading to new algorithmic solutions to minimize it. We also investigate concave formulations of this norm, corresponding to more aggressive strategies to induce orthogonality, and show how these penalties can also be used to learn sparse models with disjoint supports.

1 Introduction

Learning several models simultaneously instead of separately, a framework often referred to as multitask or transfer learning, is a powerful setting to leverage information across related but different problems [10, 22, 4, 2, 12]. In particular it has been empirically shown that when different tasks share some similarity, such as learning binding models for similar proteins [14], predicting exams score for students of different schools [2, 12] or learning models for semantically related concepts in a hierarchy [16, 8], jointly learning the different models with a multitask strategy leads to better performance. In all aforementioned examples (and many others), the underlying assumption is that different tasks share some similarity, and the different multitask strategies exploit this assumption by, e.g., imposing shared parameters estimated jointly across the tasks, or penalizing differences between the models learned in different tasks.

Alternatively, in some situations we would like to solve different tasks under the opposite assumption, namely, that the models are *different*, e.g., that they use different features or should be orthogonal to each other. This is the case for example when we want to learn unrelated tasks, such as recognizing the identity and the emotion of a person on a picture, where we know from literature that these two recognition problems depend on different and uncorrelated features of the same image [9, 19]. In structured learning such as classification in a hierarchical taxonomy, it has been proposed to learn local models at each node of

the hierarchy and to encourage the classifier at each node to be different from the classifiers at its ancestors, in order to better reflect the natural coarse-to-fine nature of the classifiers at different levels of the hierarchy [24, 13]. Several approaches have been proposed recently to learn such different models. [24] proposed to penalize a weighted ℓ_1 norm of the off-diagonal entries of the covariance matrix between the tasks, in order to promote sparsity of inner products hence orthogonality between tasks; however some extra ridge term must be added in order to make the penalty convex and amenable to efficient optimization, leading to potentially unwanted over-regularization. [19] proposed also a convex penalty to learn two groups of tasks based on orthogonal subspaces; again, due to the non-convex nature of the norm applied to inner products between vectors, an extra ridge term is needed to make the penalty convex. Finally, [13] proposed a method to learn a tree of metrics, enforcing disjoint sparsity between the different metrics. The convex penalty of [13], though, only promotes sparsity for nonnegative vectors, such as the diagonals of metric matrices, and can not easily be extended to enforce disjoint sparsity on general vectors.

In this work, we extend the work of [24] in two directions. First, we investigate generalization of the penalty proposed by [24] when we decrease its convexity, in order to make it more "aggressive" in promoting orthogonality. Our main findings can be visualized in Figure 1, which shows the level sets of penalties we consider. Starting from the strictly convex penalty of [24], corresponding to a strictly convex unit ball with singularities at matrices with orthogonal columns (left), we show that by reducing its convexity it converges to a convex atomic norm [11], whose unit ball is the convex hull of the singularities of the first ball. This shows that for particular choices of parameters the penalty of [24] is "optimal" to learn matrices with pairwise orthogonal columns, in the sense that it is the tightest convex function which is equal to the Frobenius norm on the subset of matrices that we are interested in. This observation has also algorithmic consequences: while [24] propose an optimization scheme that only works when the penalty is strictly convex, we show that the dual norm in the limit case of the atomic norm can be estimated efficiently by solving a small semidefinite program (SDP), leading to new algorithmic solutions to use this norm as regularizer in a learning problem. We also propose and investigate empirically more concave extensions of this norm in order to increase the propensity to learn matrices with orthogonal columns (right). Our second extension is to show how these penalties can be modified to learn sparse models with disjoint supports, a particular case of orthogonal models which is relevant when different tasks are know to involve different features.

2 An atomic norm to learn matrices with orthogonal columns

We consider the problem of learning a $d \times T$ matrix $W = (w_1, \ldots, w_T)$, where each column w_i is a d-dimensional vector corresponding to a task such as a linear classification model at a node of a taxonomy. We call such a matrix *scaled*

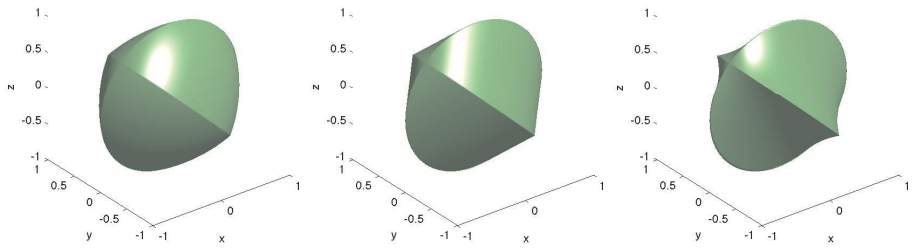

Fig. 1. Level sets of the penalty Ω_K defined in (2) for 2-by-2 symmetric matrices parametrized as $\begin{pmatrix} x & y \\ y & z \end{pmatrix}$, when $K = \begin{pmatrix} \gamma & 1 \\ 1 & \gamma \end{pmatrix}$ and we vary γ from $\gamma = 2$ (left), which corresponds to a strictly convex penalty proposed by [24], to $\gamma = 1$ (center), which is a limit case where the penalty is convex but not strictly convex and turns out to be an atomic norm (Theorem 1), and to $\gamma = 1/2$ (right), which corresponds to a non convex penalty.

orthogonal if $W^\top W$ is diagonal, i.e., if all columns of W are orthogonal to each other, and denote by $\mathcal{O}$ the set of $d \times T$ scaled orthogonal matrices. Note that this should not be confused with the stronger concept of orthogonal matrix often used in mathematics, which means that W is square and $W^\top W$ is the identity, i.e., that the columns form an orthonormal basis.

A general approach to estimate W from observations is to formulate the inference as an optimization problem:

$$\min_W f(W) + \frac{\lambda}{2}\, \Omega(W)^2\,, \tag{1}$$

where $f(W)$ is an empirical risk which measures the fit to data (e.g., variance captured in the case of dimensionality reduction, empirical error on the training set in the case of regression and classification), $\Omega(W)$ is a penalty that enforces some constraints on the solution such as sparseness or low-rankness, and $\lambda > 0$ is a parameter adjusting the tradeoff between these two objectives. When $f(W)$ and $\Omega(W)$ are convex functions, then (1) is a convex optimization problem that can often be solved efficiently and lead to a unique solution. Classical examples of penalties $\Omega(W)$ include the ℓ_1 norm to promote sparsity in W [23], the nuclear norm to learn low-rank matrices [21], and the ℓ_1/ℓ_2 norm to perform joint feature selection across tasks [17].

Suppose we know that some or all of the columns of W should be orthogonal to each other. [24] proposed an orthogonal regularizer of the form $\sum_{i,j} K_{i,j} |w_i^\top w_j|$, where $K_{i,j}$ is a nonnegative weight to enforce more or less the orthogonality between w_i and w_j. This is however not a convex function of

W, and [24] propose to define a convex penalty by adding ridge terms to this regularizer, namely:

$$\Omega_K(W)^2 = \sum_{i=1}^{T} K_{ii}||w_i||^2 + \sum_{i\neq j} K_{ij}|w_i^\top w_j|\,, \tag{2}$$

where K is an hyperparameter matrix representing structure among different models. [24] give a sufficient condition on K to ensure that (2) is convex, but there remains a lot of freedom in the choice of K.

Let us consider the case where we choose $K_{ii} = 1$ and $K_{ij} > 0$ in (2). Then we see that for scaled orthogonal matrices $W \in \mathcal{O}$ the penalty (2) boils down to the Frobenius norm:

$$\forall W \in \mathcal{O}\,, \quad \Omega_K(W)^2 = \sum_{i=1}^{T} ||w_i||^2 = ||W||_F^2\,.$$

The extra terms $K_{ij}|w_i^\top w_j|$ in (2) ensure that, in addition, the penalty is not differentiable at scaled orthogonal matrices, allowing under some conditions the recovery of such matrices when (2) is plugged into (1) [1, 11].

There are however many penalties, including (2), that are convex, singular on $\mathcal{O}$ and which equal the Frobenius norm in $\mathcal{O}$. Among them, we propose to consider the *tightest* one, namely, the atomic norm in the sense of [11] induced by the set of atoms $\mathcal{A} = \{W \in \mathcal{O} : ||W||_F = 1\}$. This norm, which we denote below by $\Omega_{\mathcal{O}}(X)$ for any $d \times T$ matrix X, can be expressed as

$$\Omega_{\mathcal{O}}(X) = \inf\left\{ \sum_{Y\in\mathcal{A}} \lambda_Y : X = \sum_{Y\in\mathcal{A}} \lambda_Y Y,\, \lambda_Y \geq 0 \right\}. \tag{3}$$

In other words, this last expression writes $\Omega_{\mathcal{O}}(X)$ as the ℓ_1 norm of the vector of coefficients λ in a decomposition of X into atoms, namely, scaled orthogonal matrices of unit Frobenius norms. Plugging (3) into (1) provides a convex problem to infer an atom, or a sparse combination of atoms. Note that, contrary to Ω_K (2), $\Omega_{\mathcal{O}}$ is always convex without technical conditions. In addition, since both norms are equal on the atoms $\mathcal{A}$, the tangent cone of $\Omega_{\mathcal{O}}$ at any scaled orthogonal matrix $W \in \mathcal{O}$ is contained in the tangent cone of Ω_K at the same point, suggesting that the recovery and inference of a scaled orthogonal matrix through the convex procedure (1) is easier with $\Omega_{\mathcal{O}}$ than with Ω_K [11].

The following result shows that, surprisingly, the norms Ω_K with adequate weights and $\Omega_{\mathcal{O}}$ coincide on matrices with two columns. This theorem is illustrated in Figure 1, where we show the unit ball of Ω_K when we change K. The ball at the center corresponds to a limit situation where Ω_K is still convex, but not strictly convex anymore. We see in this picture that the ball can equivalently be defined as the convex hull of two circles, which correspond precisely to the set of matrices with orthogonal columns and unit Frobenius norm; i.e., that Ω_K in this case is precisely the atomic norm induced by these atoms.

Theorem 1. *For any $d \geq 1$ and any $d \times 2$ matrix $W = (w_1, w_2)$, it holds that:*

$$\Omega_{\mathcal{O}}(W) = \Omega_K(W)\,, \tag{4}$$

with

$$K = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}. \tag{5}$$

Proof. Since K in (20) is entry-wise nonnegative, and since the companion matrix

$$\bar{K} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

is positive semidefinite, we know from [24, Theorem 1] that Ω_K^2 is convex in this case. Since (4) obviously holds for $W \in \mathcal{O}$, and since $\Omega_{\mathcal{O}}$ is the tightest convex function such that (4) holds on $\mathcal{O}$, we directly get that $\Omega_{\mathcal{O}}(W) \leq \Omega_K(W)$ for any $W \in \mathbb{R}^{d\times 2}$. To prove the converse inequality, it suffices to find, for any $W \in \mathbb{R}^{d\times 2}$, a decomposition of the form $W = \lambda U + (1-\lambda)V$, with $U, V \in \mathcal{O}$, $\lambda \in [0,1]$, such that $\Omega_K(U) = \Omega_K(V) = \Omega_K(W)$. Geometrically, this would mean that any point on the unit ball of Ω_K lies on a straight segment that connects two atoms on this ball, meaning that the unit ball of Ω_K is precisely the convex hull of the unit ball restricted to the atoms. The following lemma, which can be proved by direct calculation, shows that this is indeed possible by explicitly providing such a decomposition.

Lemma 1. *For any $W = (w_1, w_2) \in \mathbb{R}^{d\times 2}$, let:*

- *if $w_1^\top w_2 \geq 0$, $U = (w_1 + w_2, 0)$ and $V = \left(w_1 - \frac{w_1^\top w_2}{\| w_2 \|^2} w_2, \left(1 + \frac{w_1^\top w_2}{\| w_2 \|^2}\right) w_2\right)$,*
- *if $w_1^\top w_2 < 0$, $U = (w_1 - w_2, 0)$ and $V = \left(w_1 - \frac{w_1^\top w_2}{\| w_2 \|^2} w_2, \left(1 - \frac{w_1^\top w_2}{\| w_2 \|^2}\right) w_2\right)$,*

and let $\lambda = \frac{\left| w_1^\top w_2 \right|}{\left| w_1^\top w_2 \right| + \| w_2 \|^2}$. Then it holds that:

- *$U, V \in \mathcal{O}$,*
- *$\lambda \in [0,1]$ and $W = \lambda U + (1-\lambda)V$,*
- *$\Omega_K(W) = \Omega_K(U) = \Omega_K(V)$.*

Theorem 1 can be easily generalized (with a different set of atoms) when K is any 2-by-2 symmetric, positive semidefinite matrix with non-negative entries and with 0 as eigenvalue, corresponding to the limit case where Ω_K is convex but not strictly convex: it is then always an atomic norm. The extension of Theorem 1 to more than 2 columns, however, is not true. Atoms of $\Omega_{\mathcal{O}}$ are matrices with *all* columns orthogonal to each other, so using $\Omega_{\mathcal{O}}$ as a penalty on matrices with $T > 2$ columns may either lead to such an atom, or to a sparse linear combination of atoms, which would in general have no pair of column orthogonal to each other. The following theorem, which is a simple consequence of Theorem 1, shows that for some choices of K in the $T > 2$ case, the penalty Ω_K can be written as a sum of $\Omega_{\mathcal{O}}$ that penalizes pairs of columns.

Theorem 2. *For any $T \geq 2$, let K be a symmetric T-by-T matrix with non-negative entries and such that, for any $i = 1, \ldots, T$,*

$$\forall i = 1, \ldots, T \quad K_{ii} = \sum_{j \neq i} K_{ij} \,.$$

Then, for any $d \geq 1$ and any $d \times T$ matrix $W = (w_1, \ldots, w_T)$, it holds that:

$$\Omega_K(W) = \sum_{i<j} K_{ij} \Omega_{\mathcal{O}}((w_i, w_j)) \,,$$

where $(w_i, w_j) \in \mathbb{R}^{d \times 2}$ is the matrix with columns w_i and w_j.

Proof. Let $A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$. By Theorem 1, we know that $\Omega_A((w_i, w_j)) = \Omega_{\mathcal{O}}((w_i, w_j))$ for al $i \neq j$, therefore:

$$\begin{aligned}
\sum_{i<j} K_{ij} \Omega_{\mathcal{O}}((w_i, w_j)) &= \sum_{i<j} K_{ij} \Omega_A((w_i, w_j)) \\
&= \sum_{i<j} K_{ij} \left(||w_i||^2 + \| w_j \|^2 + 2|w_i^\top w_j| \right) \\
&= \sum_{i=1}^{T} \left(\sum_{j \neq i} K_{ij} \right) \| w_i \|^2 + \sum_{i \neq j} |w_i^\top w_j| \\
&= \Omega_K(W) \,.
\end{aligned}$$

3 The dual of the atomic norm

In this section we consider the atomic norm $\Omega_{\mathcal{O}}$ for matrices with 2 columns, and show that we can efficiently compute its dual and a subgradient of its dual by solving a 6-dimensional SDP. This can be useful to provide simple duality gaps and stopping criteria to learn with convex but not strictly convex penalties Ω_K, which are in particular not amenable to optimization with the method of [24].

Remember that the dual of a norm $\Omega(X)$ is

$$\Omega^*(X) = \sup_{Y \,:\, \Omega(Y) \leq 1} \mathbf{Tr}(X^\top Y) \,.$$

Since $\Omega_{\mathcal{O}}$ is an atomic norm induced by the atom set $\mathcal{A}$, its dual satisfies [11]:

$$\Omega^*_{\mathcal{O}}(X) = \sup_{Y \in \mathcal{A}} \mathbf{Tr}(X^\top Y) \,, \tag{6}$$

and in addition any atom $Y \in \mathcal{A}$ which achieves the maximum in (6) is a subgradient of $\Omega^*_{\mathcal{O}}$ at X. We now show that computing $\Omega^*_{\mathcal{O}}(X)$ and a subgradient can be done efficiently:

Theorem 3. *For any $d \geq 1$ and $X \in \mathbb{R}^{d\times 2}$, a solution to*

$$\Omega^*_{\mathcal{O}}(X) = \sup_{Y\in\mathcal{A}} \mathbf{Tr}(X^\top Y) \tag{7}$$

can be obtained from the solution of a SDP over matrices of size 6×6.

Proof. From the definition of $\mathcal{A}$ we can reformulate (7) as:

$$\begin{aligned}\Omega^*_{\mathcal{O}}(X) = \text{maximize } & \mathbf{Tr}(Y^\top X)\\ \text{subject to } & Y^\top Y \text{ diagonal}\\ & \|Y\|_F = 1,\end{aligned}$$

in the variable $Y \in \mathbb{R}^{d\times 2}$. Because $-Y$ is a feasible point whenever Y is, this problem is equivalent to

$$\begin{aligned}\Omega^*_{\mathcal{O}}(X)^2 = \text{maximize } & \mathbf{Tr}(Y^\top X)^2\\ \text{subject to } & Y^\top Y \text{ diagonal}\\ & \|Y\|_F = 1,\end{aligned} \tag{8}$$

which is a *non-convex* quadratic program in Y. We first reformulate this problem in "vector" terms and write $z = \mathbf{vec}(Y) \in \mathbb{R}^{2d}$, so that $z^\top = (z_1^\top, z_2^\top)$ with $z_1 = Y_1$ and $z_2 = Y_2$. Problem (8) becomes

$$\begin{aligned}\text{maximize } & (X_1^\top z_1 + X_2^\top z_2)^2\\ \text{subject to } & z_1^\top z_2 = 0\\ & \|z_1\|_2^2 + \|z_2\|_2^2 = 1,\end{aligned}$$

which is again

$$\begin{aligned}\text{maximize } & (\mathbf{vec}(X)^\top z)^2\\ \text{subject to } & z^\top \begin{pmatrix} 0 & \mathbf{I}\\ \mathbf{I} & 0\end{pmatrix} z = 0\\ & z^\top z = 1\,.\end{aligned}$$

Following the classical lifting technique derived by [20, 15], we can produce a semidefinite relaxation of this last problem by changing variables, setting $Z = zz^\top$, and dropping the implicit rank constraint on Z, to get

$$\begin{aligned}\text{maximize } & \mathbf{Tr}\left(\mathbf{vec}(X)\,\mathbf{vec}(X)^\top Z\right)\\ \text{subject to } & \mathbf{Tr}\left(\begin{pmatrix} 0 & \mathbf{I}\\ \mathbf{I} & 0\end{pmatrix} Z\right) = 0\\ & \mathbf{Tr}(Z) = 1,\ Z \succeq 0,\end{aligned} \tag{9}$$

which is a SDP in the matrix variable $Z \in \mathbf{S}_{2d}$. The quadratic convexity results of [7] (see also [3], §II.14), also known as the $\mathcal{S}$-procedure or Brickman's theorem, tells us that the optimal value of the semidefinite program (9) is equal to the optimal value of the non-convex QP in (8), and a solution Y to (8) can be constructed from an optimal solution Z of (9) (see, e.g.,, [6] App. B.3 for an explicit recursive procedure).

Problem (9) is an SDP over $2d \times 2d$ matrices, which can be prohibitive in practice as soon as d gets large. Let us now show that a simple decomposition allows to reformulate the problem as a SDP of fixed dimension 6. We can compute the QR decomposition of X written $X = QR_2$ where $Q \in \mathbb{R}^{d\times d}$ is an orthogonal matrix and $R_2 \in \mathbb{R}^{d\times 2}$ with $R_2 = (R^\top, 0)^\top$ where $R \in \mathbb{R}^{2\times 2}$ is an upper triangular matrix. This means that without loss of generality, the original problem of computing $\Omega^*(X)$ can be rewritten

$$\begin{array}{c} \text{maximize } \mathbf{Tr}(Y^T Q R_2) \\ \text{subject to } Y^T Q Q^T Y \text{ diagonal} \\ \|Q^T Y\|_F = 1\,, \end{array} \tag{10}$$

which is equivalent to

$$\begin{array}{c} \text{maximize } \mathbf{Tr}(Y^T R_2) \\ \text{subject to } Y^T Y \text{ diagonal} \\ \|Y\|_F = 1\,, \end{array}$$

in the variable $Y \in \mathbb{R}^{d\times 2}$. This means that we can always assume that X is block upper diagonal with lower block equal to zero. This program can be rewritten

$$\begin{array}{c} \text{maximize } (\mathbf{vec}(R_2)^T z)^2 \\ \text{subject to } z^T \begin{pmatrix} 0 & \mathbf{I}_d \\ \mathbf{I}_d & 0 \end{pmatrix} z = 0 \\ z^T z = 1\,, \end{array}$$

in the variable $z = \mathbf{vec}(Y) \in \mathbb{R}^{2d}$. Now notice that

$$\begin{pmatrix} 0 & \mathbf{I}_d \\ \mathbf{I}_d & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \mathbf{I}_d = (P^T \,\mathbf{diag}(-1,1) P) \otimes \mathbf{I}_d\,,$$

where $P = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}$ is an orthogonal matrix. Let us write $S = P \otimes \mathbf{I}_d$ (also an orthogonal matrix), $w = Sz$ and $b = S\,\mathbf{vec}(R_2)$, we can rewrite the QP above as

$$\begin{array}{c} \text{maximize } (\mathbf{vec}(R_2)^T S^T w)^2 \\ \text{subject to } w^T \begin{pmatrix} -\mathbf{I}_d & 0 \\ 0 & \mathbf{I}_d \end{pmatrix} w = 0 \\ w^T w = 1\,, \end{array}$$

in the variable $w \in \mathbb{R}^{2d}$. Now $b = S\,\mathbf{vec}(R_2)$ means

$$b = (P \otimes \mathbf{I}_d)\,\mathbf{vec}(R_2) = \mathbf{vec}(R_2 P)\,,$$

so if $R_2 = (R^T, 0)^T$ where $R \in \mathbb{R}^{T\times T}$ as above, then $b = \mathbf{vec}((P^T R^T, 0)^T)$ hence the b has only four nonzero coefficients at indices $J = \{1, 2, d+1, d+2\}$. This means that the QP can be reformatted as

$$\begin{array}{c} \text{maximize } w_J^T (b_J b_J^T) w_J \\ \text{subject to } w_J^T \begin{pmatrix} -\mathbf{I}_2 & 0 \\ 0 & \mathbf{I}_2 \end{pmatrix} w_J = y_1^T y_1 - y_2^T y_2 \\ w_J^T w_J + y_1^T y_1 + y_2^T y_2 = 1\,, \end{array}$$

in the variables $w_J \in \mathbb{R}^4$ and $y_1, y_2 \in \mathbb{R}^{d-2}$, where we have defined $z_1^T = (w_3, \dots, w_d)$ and $z_1^T = (w_{d+3}, \dots, w_{2d})$. By symmetry, we can assume w.l.o.g. that the coefficients of the vectors y_1 and y_2 are uniformly equal to scalars $y_1, y_2 \in \mathbb{R}$, so the last problem is equivalent to

$$\begin{aligned} &\text{maximize } w_J^T (b_J b_J^T) w_J \\ &\text{subject to } w_J^T \begin{pmatrix} -\mathbf{I}_2 & 0 \\ 0 & \mathbf{I}_2 \end{pmatrix} w_J = (d-2)y_1^2 - (d-2)y_2^2 \\ &\qquad\qquad w_J^T w_J + (d-2)y_1^2 + (d-2)y_2^2 = 1\,, \end{aligned}$$

which is now a QP of dimension 6 in the variables $w_J \in \mathbb{R}^4$ and $y_1, y_2 \in \mathbb{R}$. This last problem can then be lifted as above, to become

$$\begin{aligned} &\text{maximize } \mathbf{Tr}\, W \begin{pmatrix} b_J b_J^T & 0 \\ 0 & 0 \end{pmatrix} \\ &\text{subject to } \mathbf{Tr}\, W \begin{pmatrix} -\mathbf{I}_2 & 0 & 0 \\ 0 & \mathbf{I}_2 & 0 \\ 0 & 0 & \mathbf{diag}(-(d-2), (d-2)) \end{pmatrix} = 0 \\ &\qquad\qquad \mathbf{Tr}\, W \begin{pmatrix} \mathbf{I}_4 & 0 \\ 0 & (d-2)\mathbf{I}_2 \end{pmatrix} = 1,\ W \succeq 0, \end{aligned} \tag{11}$$

which is a semidefinite program in the variable $W \in \mathbf{S}_6$. The optimal values of programs (10) and (11) are equal and a solution to (10) can be constructed from an optimal solution to (11). Because (11) is a semidefinite program of fixed dimension 6, it can be solved efficiently *independently of the dimension d.* All we need is the QR decomposition of X which can be formed with cost $O(d)$ when $X \in \mathbb{R}^{d \times 2}$.

4 Algorithms

In order to learn with the penalty Ω_K we need to solve problems of the form

$$\min_W f(W) + \frac{\lambda}{2}\, \Omega_K(W)^2\,. \tag{12}$$

When Ω_K is strictly convex, [24] propose a regularized dual averaging (RDA) method based on subgradient descent, and show that a subgradient of $\Omega_K(W)$ in that case is given by $G = (g_1, \dots, g_t)$ where

$$g_i = K_{ii} w_i + \sum_{j \neq i} \operatorname{sign}\left(w_i^\top w_j\right) K_{ij} w_j\,, \tag{13}$$

with the convention $\operatorname{sign}(0) = 0$. When Ω_K is not strictly convex, e.g., when it is a sum of atomic norms as in Theorem 2 or when it is not even convex (as on the right-hand plot of Figure 1), the RDA methods can not be used anymore. In that case, we propose to use a classical subgradient descent scheme using the subgradient (13), and a step size decreasing with $t^{-1/2}$ where t is the iteration.

Note that [24] only prove that (13) is a valid subgradient when Ω_K is convex; we keep the same formula in the general case since Ω_K is differentiable almost everywhere. In the non-convex case, subgradient descent will converge to a stationary point, so one may run it several times with random initializations before taking the best solution. In the experiments below, we always run subgradient descent starting from the null matrix, and observed empirically that it often leads to a good solution compared to random initialization.

Let us now discuss another possible optimization scheme when K satisfies the conditions of Theorem 2, i.e., when the penalty is a linear combination of nuclear norms over pairs of columns. In that case, by Theorem 2 the optimization problem has the form:

$$\min_W f(W) + \frac{\lambda}{2} \sum_{i<j} K_{ij} \Omega_{\mathcal{O}}((w_i, w_j))^2 \,. \tag{14}$$

We can then write an equivalent dual problem amenable to optimization. Let us first consider the simple case of $T = 2$ columns, in which case (14) boils down to

$$\min_W \left\{ f(W) + \frac{\lambda}{2} \, \Omega^2_{\mathcal{O}}(W) \right\} \tag{15}$$

in the variable $W \in \mathbb{R}^{d \times 2}$. Remember that for any norm, if $h(x) = \|x\|^2/2$ then the Fenchel dual of h is $h^*(y) = \|y\|_*^2/2$ [6, §3.3.1]). Then [5, Th. 3.3.5] shows that the dual of (15) is written

$$\sup_Z \left\{ -f^*(Z) - \frac{1}{2\lambda} \left(\Omega^*_{\mathcal{O}}(Z)\right)^2 \right\} \tag{16}$$

in the variable $Z \in \mathbb{R}^{d \times 2}$. Under mild technical conditions, the optimal values of both problems are equal. Back to the general case (14), note that the conjugate of the function $\Omega_{\mathcal{O}}((w_i, w_j))$, which we write $\tilde{\Omega}^*_{ij}(W)$, is given by

$$\tilde{\Omega}^*_{ij}(W) = \begin{cases} \Omega^*_{\mathcal{O}}((W_i, W_j)) & \text{if } W_l = 0 \text{ for } l \neq i, j \\ +\infty & \text{otherwise.} \end{cases}$$

Then, using the following inf-convolution result [18, Th. 16.4]:

$$(f_1 + \ldots + f_s)^*(y) = \inf_{y_1, \ldots, y_s} \{ f_1^*(y_1) + \ldots + f_s^*(y_s) : y_1 + \ldots + y_s = y \},$$

we obtain that the Fenchel dual of problem (14) is written

$$\sup_Z \left\{ -f \left(\sum_{i<j} Z_{ij} \right) - \sum_{i<j} \frac{1}{2\lambda K_{ij}} \tilde{\Omega}^*_{ij}(Z_k)^2 \right\} \tag{17}$$

in the variables $(Z_{ij})_{i<j} \in \mathbb{R}^{d \times T}$. Note that the definitions of $\tilde{\Omega}^*_{ij}$ mean that each Z_{ij} only has two nonzero columns at positions i and j. Now, note that by Theorem 3, the function to be optimized in (17) can be efficiently estimated and a subgradient can be computed. Any value of (17) provides a lower bound to (14), thus giving a duality gap that can be used to monitor convergence of the subgradient descent method.

5 Learning disjoint supports

An interesting particular case of learning orthogonal vectors is the situation where we seek sparse vectors with disjoint supports. In this section we briefly discuss how Ω_K can help in this situation, too. For simplicity we only discuss the case of $T = 2$ vectors, an extension to the general case being straightforward. The matrix $W \in \mathbb{R}^{d\times 2}$ has columns with complementary supports if, for $i = 1, \dots, d$,

$$W_{1,i} \neq 0 \implies W_{2,i} = 0 \text{ and } W_{2,i} \neq 0 \implies W_{1,i} = 0\,,$$

or in other words $W_1 \circ W_2 = 0$ where $\circ$ denotes the Hadamard (entrywise) product of matrices. If we denote by $|W|$ the matrix whose entries are the absolute values of the entries of W, then we further observe that $|W_1 \circ W_2| = |W_1| \circ |W_2|$, so $W_1 \circ W_2 = 0$ if and only if $|W_1| \circ |W_2| = 0$. Interestingly, if $V \in \mathbb{R}^{d\times 2}$ is a matrix with non-negative entries, then $V_1 \circ V_2 = 0$ is equivalent to $V_1^T V_2 = 0$; this shows that W has columns with complementary supports if and only if $|W_1|$ and $|W_2|$ are orthogonal.

This suggest a general way to learn a matrix with disjoint supports, by solving a problem of the form:

$$\min_W f(W) + \frac{\lambda}{2}\,\Omega_K(|W|)^2\,, \tag{18}$$

where Ω_K is a penalty that induces orthogonality among columns. To solve (18), we introduce a non-negative matrix V such that $-V \leq W \leq V$ (where $\leq$ refers to elementwise comparisons), and solve the following problem:

$$\min_{-V\leq W\leq V} f(W) + \frac{\lambda}{2}\,\Omega_K(V)^2\,. \tag{19}$$

At the optimum of (19), we have $V = |W|$ which shows that (19) is indeed equivalent to (18). Since a subgradient of (19) in (V, W) can easily be computed, we propose to solve a(19) by a projected subgradient scheme, where at each iteration we update V and W with a move along a subgradient, and then project the new point to the constraint set $-V \leq W \leq V$ and $V \geq 0$.

6 Experiments

In this section, we present numerical experiments on two simulated datasets. We benchmark the following methods:

- Xiao: this is the method described in [24] where we solve (1) with the penalty (2). We consider both with convex and non-convex versions, by changing the matrix K in (2).
- Disjoint Supports: this is the approach where we solve (18), with non-convex and convex versions.
- Ridge Regression: this standard method corresponds to learning the tasks independently by ridge regression.

– LASSO: this is the classical approach inducing sparsity over all tasks, without sharing information across the tasks.

In all experiments involving Ω_K, we consider a symmetric matrix K parametrized by its diagonal value γ,

$$K = \begin{pmatrix} \gamma & & 1 \\ & \ddots & \\ 1 & & \gamma \end{pmatrix} . \tag{20}$$

Based on the conditions for the convexity of Ω_K studied by [24], we control the convexity of Ω_K used in the Xiao and Disjoint Supports approaches with the following rule on γ:

– $\gamma > T - 1$ leads to strictly convex Ω_K function, as described in [24],
– $\gamma = T-1$ is the the limit case where Ω_K satisfies the conditions of Theorem 2, i.e., where it is a sum of atomic norms over pairs of columns:

$$\Omega_K(W) = \sum_{i<j} \Omega_{\mathcal{O}}((w_i, w_j)) , \tag{21}$$

– $\gamma < T - 1$ corresponds to the case where Ω_K is not convex.

We test the different methods on regression problem where, given a matrix of covariates $X \in \mathbb{R}^{n\times d}$ and a matrix of T response variables $Y \in \mathbb{R}^{n\times T}$, we seek to minimize the squared error $f(W) = \|Y - XW\|^2$.

6.1 The effect of convexity

We use simulated data to test whether theoretical differences between $\Omega_K, \Omega_{\mathcal{O}}$ and concave formulations have an impact on analytical performances. In particular, by playing with γ in (20), we investigate to what extent the convexity constraint imposed by [24] is restrictive in terms of performance.

For that purpose, we randomly generate models W consisting of $T = 10$ tasks in $d = 10$ dimensions, such that all tasks are orthogonal to each other. The training set X_{train} is composed of $n = 50$ instances, each element of X_{train} being sampled from a normal distribution $\mathcal{N}(0, 1)$. We simulate the response variable $Y_{train} \in \mathbb{R}^{n\times T}$ according to $Y_{train} = X_{train}W + \epsilon$, where ϵ is a noise matrix of i.i.d. centered Gaussian variables with variance σ^2. We estimate the performance of each model on a test set of 1000 samples generated similarly. We also measure how orthogonal the models are, by the mean absolute difference between the angle between two columns of W and $\pi/2$. For each value of γ we estimate the Xiao model with different regularization parameters λ over a grid of 21 values regularly spaced after log transform; the grid was set to ensure that it covered good parameters for all methods. For each γ, we report the performance of the best λ in terms of test MSE. We repeat the full procedure 100 times and report the average results over the 100 repeats.

Figure 2 shows the performance of the methods in terms of test error (top), and in terms of how far the models learned are from orthogonal models (bottom).

On each plot, the horizontal axis is the γ parameter on the diagonal of K defined in (20), and the vertical dotted line corresponds to the atomic norm (21) and is the transition from convex (to its right) to non-convex (to its left). From left to right, we show results corresponding to increasing noise in the response variable, with the variance of ϵ set respectively to $1, 2.5$ and 4. We see that in the small noise regime (left), non-convex formulations perform better while with high noise (right), the convex formulations are more adapted. Inbetween (middle), the best performance is reached for slightly non-convex penalties. In all cases, the models learned are similar in terms of how non-orthogonal they are; we see that non-convex formulations lead to significantly more orthogonal models than convex formulations. Overall, these results suggest that restricting ourselves to strictly convex penalties may be restrictive and sub-optimal in some cases; they show that non-convex penalties can allow to learn more orthogonal models with better performance.

6.2 Regression with disjoint supports

As a second proof of concept, we check the relevance of the formulation presented in Section 5 to jointly learn linear models with disjoint support. For that purpose, we simulate data as in Section 6.1, with the additional constraint that the columns of W are orthogonal and have disjoint supports. Since $d = T = 10$, this means that W is simply diagonal. We fix the noise level at $\sigma^2 = 1$, and simulate training sets of increasing size between 10 and 50 samples, repeating the full procedure 100 times. We compare four methods: (i) the Xiao model with varying parameter γ according to (20), leading to orthogonal but non-sparse vectors, (ii) our new method (18) again with convex and non-convex formulations by varying γ in (20), (iii) a baseline ridge regression model and (iv) a LASSO regression model leading to sparse but not necessarily orthogonal vectors. For each model, a 5-fold cross-validation is performed on the training set to select an optimal regularization parameter, which is then used to train the model on the full training set before doing a prediction on an independent test set. We assess the performance of each method on the test set in terms of accuracy (measured by the MSE), and in terms of disjoint support recovery, measured as the proportion of features which are correctly selected in a single column of W.

The results are shown in Figure 3, where for sake of clarity we only report the results of Xiao and Disjoint Supports for the optimal diagonal value γ, which in both cases is equal to 0.1, corresponding to a very non-convex penalty. In terms of performance, we see that Xiao is a bit better than Ridge regression for $n = 50$ training point, which is coherent with the observation made in Section 6.1 in the small-noise regime, although for less than 30 samples Ridge regression is better. Both methods are outperformed by LASSO, which in this case benefits from the very sparse structure of W. Interestingly, the new Disjoint Support model significantly outperforms all other methods for all training set sizes (P-value $< 10^{-3}$). As for the ability of different methods to correctly recover the disjoint supports, we see that Disjoint Supports shows increasing support recovery score for large training set size, and outperforms LASSO which induces global sparsity

but is not able to affect features to an unique column. Ridge Regression and Xiao are not shown because they do not achieve any sparsity in the model they learn. In summary, this simulation shows that the Disjoint Supports model has the potential to outperform other methods when the model to learn is sparse with disjoint supports.

7 Conclusion

We have extended the work of [24] in two directions: on the one hand, we have investigated the possibility to work with non strictly convex or non convex formulations, leading to more agressive control of model orthogonality, and on the other hand we have shown how models to learn orthogonal columns can be extended to learn sparse models with disjoint supports. In the two-columns case, we have proved that the penalty of [24] is an atomic norm derived from the set of scaled orthogonal matrices, and for the general case $T > 2$ we have shown that for suitable choices of parameters it can be written as a linear combination of atomic norms applied to pairs of columns. In terms of algorithms, the RDA algorithm proposed by [25] is only suitable to solve the problem (12) in the strictly convex case, and we have shown that in the limit case where Ω_K is convex but not strictly convex we can solve iteratively with a series of 6-dimensional SDP. Our simulations show that considering non-convex versions of the penalty can be relevant, in particular for small noise regime. Interestingly, we observed that non-convex formulations lead to more orthogonal models than convex formulations, and that the Disjoint Support model significantly outperformed all other models when the disjoint support hypothesis was met. In the future, we plan to investigate the relevance of this approaches with more structured matrices K, such as the ones used for hierarchical classifications [24] or learning groups of models [19].

References

1. Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2011.
2. Bakker, B. and Heskes, T. Task clustering and gating for bayesian multitask learning. *J. Mach. Learn. Res.*, 4:83–99, 2003.
3. Barvinok, A. *A Course in Convexity*. American Mathematical Society, 2002.
4. Baxter, J. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.
5. Borwein, J.M. and Lewis, A.S. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. Cms Books in Mathematics Series. Springer Verlag, 2000.
6. Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.
7. Brickman, L. On the field of values of a matrix. *Proceedings of the American Mathematical Society*, pp. 61–66, 1961.

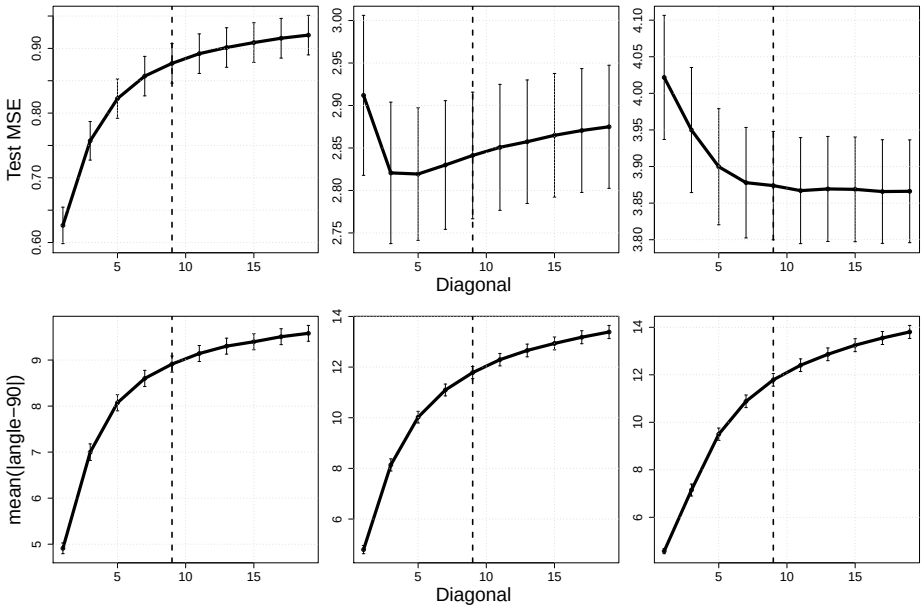

Fig. 2. Test MSE (top) and deviation from pairwise orthogonality (bottom) as a function of the convexity parameter γ, from low to high noise regimes (from left to right: $\sigma^2 \in \{1, 2.5, 4\}$). On each plot, the horizontal axis is the γ parameter on the diagonal of K defined in (20). The vertical dotted line corresponds to the atomic norm (21).

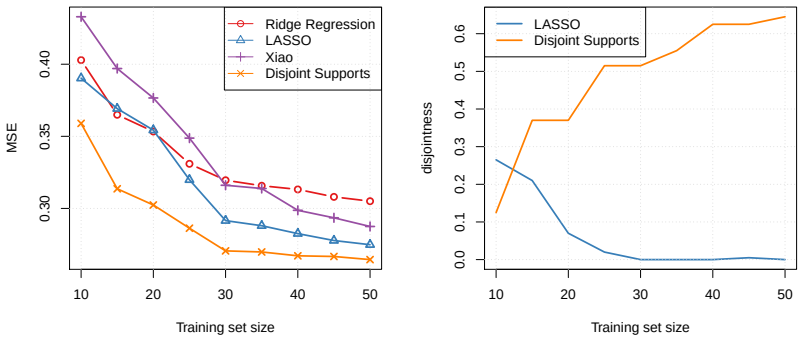

Fig. 3. Sparse regression with disjoint supports. Test MSE for training set of increasing size (left), and proportion of correctly affected features (right). Ridge regression and Xiao are not shown on the right plot because they are not sparse.

8. Cai, L. and Hofmann, T. Hierarchical document categorization with support vector machines. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pp. 78–87, New York, NY, USA, 2004. ACM.
9. Calder, A. J., Burton, A. M., Miller, P., Young, A. W., and Akamatsu, S. A principal component analysis of facial expressions. *Vision Res*, 41(9):1179–1208, Apr 2001.
10. Caruana, R. Multitask learning. *Mach. Learn.*, 28(1):41–75, 1997.
11. Chandrasekaran, V., Recht, B., Parrilo, P. A., and Willsky, A. S. The convex geometry of linear inverse problems. *Found. Comput. Math.*, 12(6):805–849, 2012.
12. Evgeniou, T., Micchelli, C., and Pontil, M. Learning multiple tasks with kernel methods. *J. Mach. Learn. Res.*, 6:615–637, 2005.
13. Hwang, S. J. J., Grauman, K., and Sha, F. Learning a tree of metrics with disjoint visual features. In Shawe-Taylor, J., Zemel, R.S., Bartlett, P., Pereira, F.C.N., and Weinberger, K.Q. (eds.), *Adv. Neural. Inform. Process Syst. 24*, pp. 621–629. 2011.
14. Jacob, L. and Vert, J.-P. Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics*, 24(19):2149–2156, 2008.
15. Lovász, L. and Schrijver, A. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.
16. McCallum, A., Rosenfeld, R., Mitchell, T. M., and Ng, A. Y. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 359–367, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
17. Obozinski, G., Taskar, B., and Jordan, M.I. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, 2010.
18. Rockafellar, R. T. *Convex Analysis*. Princeton University Press., Princeton., 1970.
19. Romera-Paredes, B., Argyriou, A., Berthouze, N., and Pontil, M. Exploiting unrelated tasks in multi-task learning. *J. Mach. Learn. Res. - Proceedings Track*, 22: 951–959, 2012.
20. Shor, N.Z. Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, 25:1–11, 1987.
21. Srebro, N., Rennie, J. D. M., and Jaakkola, T. S. Maximum-margin matrix factorization. In Saul, L. K., Weiss, Y., and Bottou, L. (eds.), *Adv. Neural. Inform. Process Syst. 17*, pp. 1329–1336, Cambridge, MA, 2005. MIT Press.
22. Thrun, S. and Pratt, L. (eds.). *Learning to learn*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.
23. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B*, 58(1):267–288, 1996.
24. Xiao, L., Zhou, D., and Wu, M. Hierarchical classification via orthogonal transfer. In Getoor, L. and Scheffer, T. (eds.), *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011.*, pp. 801–808. Omnipress, 2011.
25. Xiao, L. Dual averaging methods for regularized stochastic learning and online optimization. *J. Mach. Learn. Res.*, 9999:2543–2596, 2010.